Usable Verifiable Secrecy-Preserving E-Voting

Oksana Kulyk[1], Jonas Ludwig[2], Melanie Volkamer[2], Reto E. Koenig[3], and Philipp Locher[3]

[1] IT University of Copenhagen, Rued Langgaards Vej 7, DK-2300 Copenhagen S
[2] Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen
[3] Bern University of Applied Sciences, Quellgasse 21, 2501 Biel

Abstract. In this paper we propose the usage of QR-Codes to enable usable verifiable e-voting schemes based on code voting. The idea – from a voter's perspective – is to combine code voting proposed by Chaum with the cast-as-intended verification mechanism used e.g. in Switzerland (using a personal initialization code, return codes per option, a confirmation code and a finalisation code); while all codes to be entered into the e-voting system by voters are available as QR-Code (i.e. one personalised QR voting code per voting option and one personal confirmation QR-Code). We conduct a user study to evaluate the usability and user experience of such an approach: both the code sheets and the election webpage are based on usability research in this area but adopted for our idea. As our proposal performs good wrt. usability, we discuss how such usable front-ends enable more secure e-voting systems in respect to end-to-end verifiability and vote secrecy.

1 Introduction

Developing e-voting systems with high level of security using non trustworthy voting clients (i.e. malware infected voting machine or personal computer equipment in case of remote e-voting) is very delicate and existing proposals come with severe usability challenges. In the most recent past, this challenge – both in academia as well as in conducting e-elections – was addressed with respect to vote integrity. Several cast-as-intended verification methods have been proposed to allow voters themselves to perform checks to verify that their vote has not been manipulated; in particular that it has not been manipulated by their compromised voting component. There are different schemes depending on the accepted trust assumptions: e.g. methods involving return codes and rely on trust in the postal service e.g. as used in Switzerland [1] or methods involving a second hardware device and rely on trust in the independence of this device from the device used to cast the vote, e.g. as used in Estonia [32]. The proposed and applied cast-as-intended verification methods require additional involvement from the voter. This involvement comes with usability challenges addressed in various user centered research investigations and user studies – see e.g. [4, 8]. One well studied cast-as-intended verification method is the one based on *return codes*

as it is used by the Swiss Post voting system in Switzerland. User studies and improved voting materials and interfaces were studied e.g. in [17,22]. In particular with the improvements of [17, 22], this method is – both from a security (wrt. to integrity) and a usability point of view – very promising. From a usability point of view the most challenging and error prone task is that voters need to manually enter the *initialisation code* and the *confirmation code* provided on the voting material (the *personal paper code sheet*) they receive via postal service.

This method (like other cast-as-intended verification methods) has one major drawback: untrustworthy voting clients can violate vote secrecy. To address vote secrecy although the voting client is not trustworthy, the *code voting* approach has initially been coined by David Chaum in 2001 [9] and was re-addressed in [28]. The idea of code voting is that voters enter a secret, individual code that corresponds to their desired option on their personal paper code sheet. However, combining both approaches, i.e. the return code approach as discussed in [17, 22] with code voting seems not worth considering as the number of error-prone voter tasks increases. Thus, researchers may not consider developing a corresponding voting protocol as it is likely to not be usable.

But what if – and this is our main idea – the codes are included in QR-Codes. Thereby, the task of entering codes becomes less error-prone and the number of interactions can be reduced as the QR-Code can contain more than one of the originally needed codes. We show how voting material could look like for such voting schemes and that it enables usable vote casting and usable verifying. Thereby, we enable new (more robust) types of usable verifiable code voting schemes and invite the community to propose corresponding secure voting protocols using the possibility to provide cryptographic data, even e.g. zero knowledge proofs, in the QR-Codes. Thus, our contributions are as follows:

- An extension of the code sheet and voting webpage of [17] to enable QR-Code based voting codes and confirmation codes.
- A user study to evaluate the *usability* of our code sheet and voting webpage.
- A discussion of opportunities for more *secure* voting protocols with our idea.

The approach that we take differs from how usability concerns are commonly incorporated into the design of secure voting systems: So far, many voting protocols were proposed and then invite the community to design voting material and election webpages being as usable as possible with the given protocol. With this paper, we provide usable front-ends for a verifiable secrecy-preserving e-voting system and invite the community to propose adequate protocols.

2 Background and Related Work

In this section, we first explain the concepts necessary to understand the verifiable code voting ceremony considered in this paper. Afterwards, we discuss related work, i.e. research on the usability of verifiable voting as well as on code voting.

2.1 Background

Verifiable Voting in General. Verifiability in the context of e-voting is no issue, as long as voter privacy is not a requirement. Every voter can verify using public data, if their intention is reflected in the final tally. This can be as simple as checking if the voter's name is alongside the clear-text vote and the sum of all votes match the tally. If, however, voter privacy is a requirement, a more complex process has to be established in order to provide vote secrecy and verifiability at the same time. In this case, verifiability splits in two parts, i.e. individual verifiability and universal verifiability. Universal verifiability can be delegated to any public entity and provides strong proof whether the final tally has been correctly derived from the recorded votes. Individual verifiability on the other hand can only be conducted by the individual voters themselves as only they know their true intention. Individual verifiability provides a strong proof to the verifying voter whether their intention is contained in the cast ballot (*cast-as-intended*) and if the recorded ballot corresponds to the cast ballot (*recorded-as-cast*). This way, the verification is complete, from the voter's intention to the final tally, and hence is called end-to-end verification.

Security Model. For an end-to-end verifiable e-voting system offering vote secrecy, its soundness is based on trust and computational intractability assumptions. The soundness of the end-to-end verifiable voting scheme is strengthened if it operates within minimal trust assumptions. This includes that no unbeknownst manipulation is possible while a minimal defined subset of entities is working honestly, whereas entities can be devices and people. The scheme should also not provide any single entity the capability to break secrecy. However, many proposals rely on the trustworthiness of the voting device when it comes to vote secrecy.

Verifiable Voting using Return/Confirmation/Finalisation Codes. There are various approaches with different trust assumptions. Our paper focuses on the approach used in Switzerland and as required in [1]: Voters receive their individual code sheet via postal service, containing one initialisation code, return codes for each voting option, one confirmation code, and one finalisation code. Voters enter their initialisation code on the election webpage and then select their voting option using the election webpage. Afterwards, they receive a return code which voters are supposed to compare with the one next to their voting option on the code sheet. If the return code is correct, the voter confirms its correctness by entering the confirmation code. If the return code is incorrect, the voter is supposed to vote via an alternative voting channel (postal or in person). Finally, voters receive a finalisation code which should match the one on their code sheet, as an assurance that their ballot has been recorded. The return code would be enough to verify, however, from an organisational perspective it is recommended to have the additional two steps and codes respectively in order to have a chance to react to complaining voters. According to the requirements in [1] and the voting material in [17], the initialisation and the confirmation code are very long (more than 20 digits). The finalisation code should consist of 8 digits (0-9) and the return code of 4 digits.

Usability Considerations. In terms of the ability to cast a vote the most error-prone task is to enter the initialisation code and then the confirmation code.

Security Model. According to the general idea of verifiable voting, no single entity should be able to break vote integrity. In particular, the voting client is considered untrustworthy. Hence, the voting scheme must ensure detection of any malicious vote manipulation. However, it is worth mentioning that a malicious online collusion of the voting client with the printer used to print the voting material or the postal service could break vote integrity. Hence, it is usually recommended to operate the printer offline. This way, the trust assumption remains that the postal service is working highly distributed and thus is more difficult to abuse for large scale attacks.

Code Voting. Code voting has one goal: The human at the far end of the e-voting system shall be enabled to provide a vote in privacy even though its voting client is controlled by malware targeting vote secrecy. The general idea from the voter's perspective is as follows: The voter is provided with a unique code sheet via a trusted secure channel and is supposed to enter the voting code representing their chosen option via the insecure voting client.

Usability Considerations. Very short voting codes may be usable. However, they are problematic from a security and operational perspective. Namely, the minimum length of the unique voting code grows fast as it depends on the size of the electorate times the number of voting options. As such, in many elections with larger electorate or with larger number of voting options, four digits are insufficient and longer codes are required in order to discriminate each eligible vote cast. This problem usually is addressed by more complex system settings, multiplexing different code semantics, e.g. voting card identifier and voting code. From a usability perspective this results in either entering a longer code per voting option or in entering multiple codes (e.g. voting-card number and voting option).

Security Model. Only a minimum subset of entities should be required to work truly honest in order to guarantee vote secrecy. In particular, the voting client is assumed not being trustworthy in any aspect. Again, the voting scheme must ensure vote privacy at this point. This is where Code Voting is at its best. However, as already mentioned in section 2.1 the printer used to print the voting material or the postal service could break vote secrecy together with the voting client.[4] Furthermore, a malicious printer could manipulate the QR codes leading to a variety of attacks, such as leading the voter to a malicious website. Our security model therefore assumes that the printed material is trustworthy, which can be ensured e.g. via proper audits before the materials are distributed to the voters.

Extending code voting. Note that there exist code voting proposals such as [9, 13] which provide somewhat verifiability. Somewhat, because they only address the (potentially) untrustworthy voting client while the overall soundness

[4] Even if a malicious printer operates offline, it can insert subliminal messages for malicious voting clients.

relies on too many trust assumptions at the server-side. The proposal by Rui et. al [14] is end-to-end verifiable and uses code voting but the election server must be ultimately trusted for vote secrecy. This trust setting is not acceptable within our security model. The proposal by Neumann et. al [24] lacks robustness as the authorities are oblivious to any malicious voter attacking the system during election phase. By sending different codes to the individual authorities, this rather simple attack results in inconsistent counting results.

2.2 Related Work

In this subsection, we review related research, i.e. user studies on verifiable electronic voting systems: researchers have explored various human factor related dimensions of verifiable electronic voting systems, including research on voters' mental models regarding verifiability in (electronic) voting and usability of verifiable voting systems.

A number of studies have explored voters' *verification-related mental models* [25, 26, 29] and revealed a number of factors that would potentially prevent voters from verifying, such as a lack of verification-related knowledge, effort required to verify, and verification-related misconceptions. These factors need to be addressed when introducing verifiable electronic voting systems.

Other human factor related electronic voting research focused on *usability of verifiable electronic voting systems*. As such, a number of them evaluated user experience and voter satisfaction in these system, e.g. [10, 11, 16, 18, 19, 27, 33]. While some studies reported high users satisfaction scores related to the voting activity, others uncovered usability issues. In particular, a study into usability issues of the Norwegian Internet voting system [11], which relies on a code-based verification, identified a lack of understandability wrt. the different codes used by the system and needed to cast / verify a vote. Distler *et al.* [10] investigated the user experience of the Selene voting system, which uses code-based verification. They reported participants feeling less secure *after* verifying than before.

Other works measure the effectiveness of verification, e.g. for Prêt à Voter and Scantegrity II in [2,3], for BingoVote in [6], for StarVote in [4], for EasyVote in [8], for Helios Internet voting system in [2, 15, 20, 31]. Some reported high rates of verification effectiveness [4, 8], others reported several issues [2, 3, 6, 20] including verification misconceptions, which resulted in participants being unable to verify their votes successfully. Note that some of these schemes were developed using a human-centered security approach. Therefore the high effectiveness rate is not too surprising. It shows once more, how important it is to take this approach when developing complex systems such as verifiable electronic voting systems. Several of such works focused on the effectiveness of code-based verification, for Estonian voting system in [12], for the Swiss voting system in [17] and for a mock system with code-based verification in [23]. In particular, the results in [12, 17] show that voters have difficulties with detecting manipulations if the adversary manages to tamper with the flow of the verification process, e.g. by removing the verification code and all mentions of it from the user interface of the voting wegpage. Marky et al. [22] furthermore propose an improvement towards the

interface of the Swiss voting system that uses code-based verification, showing high verification effectiveness as the result of the improvement.

Most relevant for our current work is the work by Kulyk et al. [17], which proposed and evaluated modifications towards the Swiss voting system [1], showing that while a high percentage of participants were not able to detect vote manipulations introduced in the study using the original system, manipulation detection was improved in the modified version.

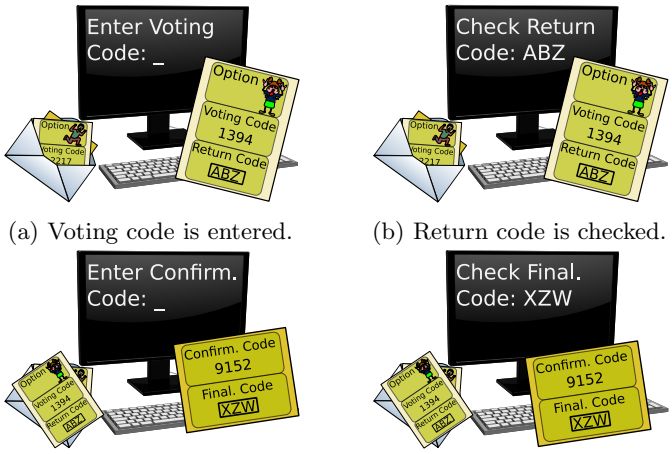
Furthermore, there is work on the usability of *code voting*: Marky et al. [21] investigated the usability of code voting, comparing different code modalities. While the authors in [21] only used QR-Codes to include 8-digit voting codes, they showed that using QR-Codes to enter the voting code is perceived as more usable than manually entering the 8-digit voting code. Thus, their results serve as basis for our research. The usability and acceptance of code voting as well as code-based verification including voting codes was also evaluated in [16]. This study reported that participants were more willing to use a system with the highest security assurance in a real-world election, even if it was less usable, which is likely given the complexity of entering and comparing different codes.

3 Voting Ceremony and Voters' Voting Material

The goal of this section is to introduce and explain our design process as well as our design decisions. Note that for this paper, our focus is on typical Swiss voting events, e.g., popular initiatives, referenda, i.e. one question with four options: yes/no/invalid/abstain. We discuss other election types in the discussion section.

3.1 Combining Code Voting with Verifiable Voting using Return/Confirmation/Finalisation Codes

For this paper, we consider a combination of code voting with individual verifiability proposals using return codes, confirmation codes and finalisation codes as used in Switzerland [1]. For such a protocol the voting ceremony is likely to be as follows: voters receive their individual voting material via postal service. Voters choose their desired option and their corresponding *voting code*. Voters enter this code in the voting equipment as depicted in fig. 1a. The device sends that code to the server side of the voting system. The server side responds to that voting code by sending back an according *return code*, depicted in fig. 1b. Voters can then verify by comparing the return code with the one that is provided for that specific option. In case the codes do not match, voters stop the voting process and report to the election management board (EMB). If, however, the return codes match, voters confirm the correct ballot casting by entering the *confirmation code* (see fig. 1c). The server side then acts again by returning the *finalization code* (see fig. 1d). If the code is wrong, voters report to the EMB. If it is correct, the vote casting process is finished and voters can be ensured that their vote has been cast-as-intended and recoreded-as-cast.



(a) Voting code is entered. (b) Return code is checked.

(c) Confirmation code is entered. (d) Finalization code is checked.

Fig. 1: Abstract Voting Ceremony

3.2 Design Decisions

Usage of QR-Codes and Smartphones. As described in the usability part of section 2.1 the length of the voting codes (or the need to have a voting card identifier plus voting codes) to enter directly depends on the size of the electorate times the number of voting options. To address these shortcomings and to enable more types of election settings for code voting, we propose that voters use their camera-equipped computer device, i.e. most likely smartphones, to cast a vote, as nowadays smartphones are capable to scan QR-Codes. Thus, both the voting codes and the confirmation code are provided as QR-Codes. Note that the initialisation code from the original Swiss approach as such is not needed anymore. The corresponding information is part of the voting code.

Election Webpage instead of App. In order to ensure cross-platform reach, we enable vote casting and verifying through a webpage which is accessible in common mobile browsers, instead of implementing it as an actual app.

Voting material from [17] and not [22]. As described in section 2.2, some research on the usability of voting materials using code-based verification exists already. In particular, the approaches from [17, 22] take the voting material used in Switzerland (for the return code approach) and improve its usability. The proposed improvements are furthermore evaluated with regard to usability as well as the ability of voters to detect election fraud. Both approaches to improve the Swiss voting material are very similar as they change the design to a more step-by-step instruction. Both studied the effectiveness wrt. voter's ability to detect various types of manipulations. Regarding simple manipulations, both improvements enabled all participants to detect them. A more advanced manipulation was only tested in [17]. While not all participants detected this advanced manipulation, the results were promising. Furthermore, the authors

conclude with proposals how to further improve the voting material to increase this detection rate. Therefore, we decided to base the voting material for our research on the proposal of [17] and try to address their proposals on how to further improve it. In particular, their results highlighted the importance of clearly defined voting steps; at the same time, it was shown that the voters primarily pay attention to the voting webpage rather than the voting sheet, which could be a problem if the voting client is compromised. For the latter reason we decided to design the webpage with only the minimal required information and controls; not distracting the voters from the voting sheet which is supposed to be their main source of instructions.

3.3 Security Considerations

In order to adopt the voting material from [17] for our purpose, i.e. for verifiable code voting, we first collected security requirements for the voting material. In particular, as the voting client is assumed to be malicious (see section 2.1), one would need to take into account the different possibilities in which a malicious smartphone could try to record everything with the camera, even without the voter granting permission. Based on that consideration, we came up with the following list of design requirements relevant to security:

- The smartphone, i.e. the camera of the smartphone, should never get knowledge about the voting options and the corresponding return codes [S0].
- The smartphone, i.e. the camera of the smartphone, should only get access to the voting code belonging to the option the voter wants to cast [S1].
- The smartphone should only have access to the confirmation code if the return code displayed on the screen of the smartphone is correct [S2].
- The smartphone should only have access to the finalisation code from the voting material after it is displayed on the screen of the smartphone [S3].

The aforementioned requirements, are addressed by the following proposals:

- There is one voting card per voting option. It shows on one side the option and the corresponding return code; and on the opposite side the corresponding voting code. Thus, in total there are four different voting cards for the election type we consider. This is required to address [S0, S1].
- The instructions in the voting material start with selecting the voting card and returning the voting cards not needed back into the envelope. Note that the election webpage has not yet been contacted nor were voters instructed to use their smartphone. This is required to address [S1].
- The instructions in the voting material afterwards state that voters should place the selected voting card on a specific area in the voting material while having the voting code visible but not the return code. Note that the voters were still *not* instructed to use their smartphone or even open the election webpage. This is required to address [S1].
- The voting material is delivered in form of a leaflet – meaning that in the inner part of the leaflet voters find the first instructions but only until the

step in which they should check the return code. They are only asked to continue to the next and last page if the return code matches. This is required to address [S2].
– The voting material should hide the finalisation code under a scratch field. This is required to address [S3] as the finalisation code and the confirmation code are on the same side of the page.

3.4 Final Voting Material

The voting material and the election webpage have been developed and iteratively improved through feedback. The final version of the voting material is depicted in fig. 2 (voting cards). [17] also includes a proposal for how to improve the election webpage. However, as we wanted to display it only on a mobile device and most likely a smartphone, we decided to keep it as simple as possible and thus reducing any information on the election webpage to the bare minimum. Another reason for such a minimalistic approach is derived from the findings from previous research [17], namely, the need to ensure that the voters follow the instructions on their trustworthy voting materials and are not distracted by instructions on the untrustworthy voting wegpage that might be manipulated by the attacker. The final version of the election webpage interfaces is depicted in fig. 4 and the voting card in fig. 3.

4 User Study

The purpose of the user study is to evaluate our proposed design for the voting materials. We want to answer the following research questions:

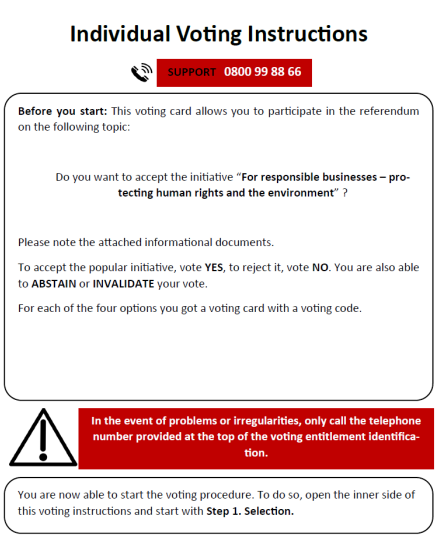
Effectiveness: Can voters successfully *cast* and *verify* their vote using the proposed system?
Satisfaction: What is the mean System Usability Scale (SUS) [5] score of the system?

In addition to these questions, we apply the modular User Experience Questionnaire [30] in order to measure the participants' impression of the proposed system wrt. *perceived efficiency*[5], *perceived perspicuity*, *perceived dependability*, and *trust*. We furthermore aimed to collect qualitative user feedback, in order to identify problems and potential improvements.

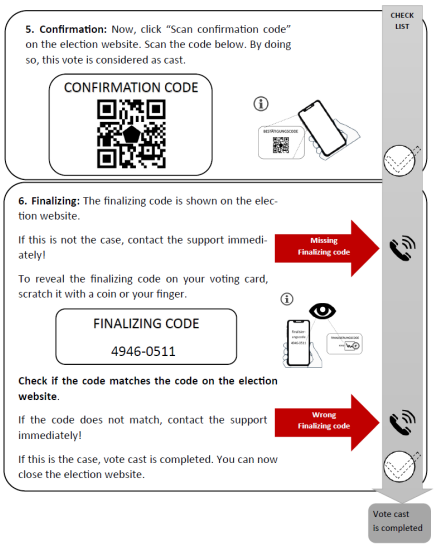
4.1 Study procedure

Before the actual study, participants received the voting materials as well as supplementary materials either via postal service or if possible in person from one

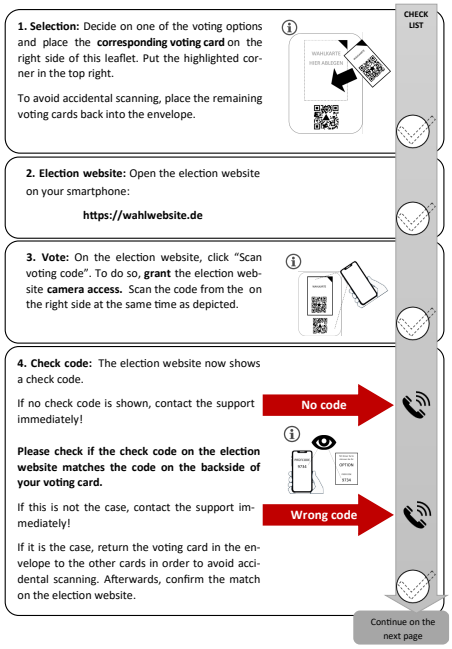
[5] Note thatas opposed to efficiency as one of the three standard usability criteria, the UEQ measures the subjective impression of whether the system feels efficient to the user, as opposed to an objective measuring of e.g. time spent on using the system



(a) front



(c) back



(b) inner

Fig. 2: Voting instructions

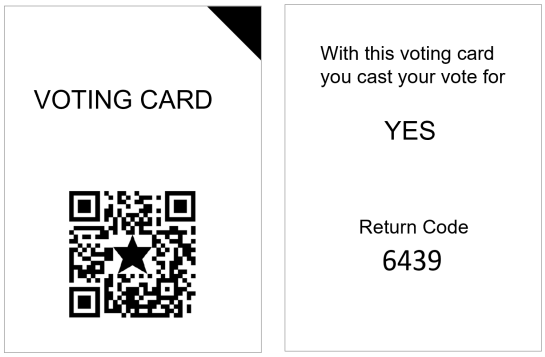
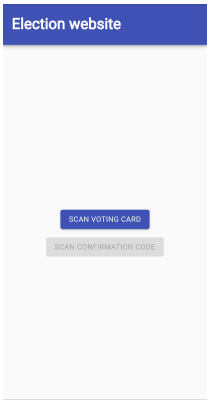
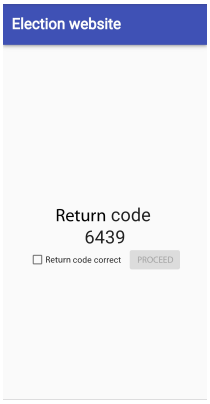


Fig. 3: Voting Card (front and back side)

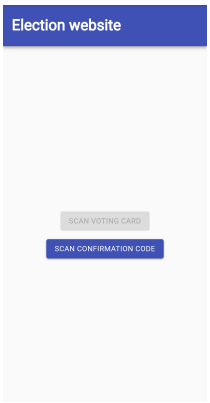


(a) Step 2

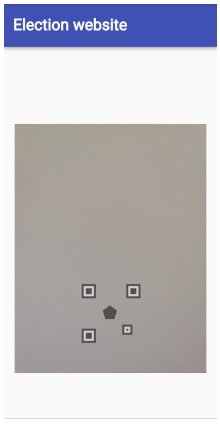
(b) Step 3



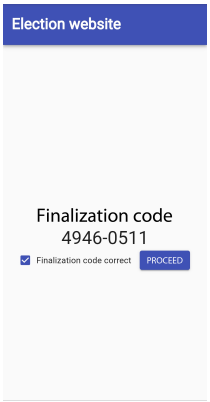
(c) Step 4



(d) Step 5a)



(e) Step 5b)



(f) Step 6

Fig. 4: Voting webpage

of the authors. The supplementary materials include a description of the purpose of the study and the study procedure as well as a role card. In the role card, participants were told to imagine that they are voters living in Bern who want to cast their vote with regards to a mock popular initiative "For responsible business in protection of people and the environment", with "yes/no/invalid/abstain" as available voting options. For the sake of the study, the participants were asked in the role card to cast their vote for a "yes" option. The participants were given a time frame of nine days during which they could cast a vote remotely and participate in the user study. For this study, the prototype webpage was developed and hosted on the "2021.wahlwebseite.de"domain and the support phone number on the voting material was the mobile phone number of one of the authors. Once participants cast their vote, the webpage provided a link to the SoSciSurvey survey platform to answer some questions. Note thatno data was collected by the webpage. It was only counted whether the support hotline was called and whether it was possible to solve their issue.

On the survey page, participants were presented with a consent form for collecting their data within this survey. Once they agreed, they were asked questions on whether they were able to complete vote casting and whether they contacted the support before or during the process. Afterwards, the participants were presented with questions from the SUS and UEQ questionnaires, asked to elaborate on any problems they had experienced, what they liked and disliked about the voting materials and the election webpage, and whether they had any improvement suggestions. The survey ended with questions on demographic variables (such as age, gender, education), previous experience with Internet voting the participant might have had and information about the device (browser, OS) that the participant used for voting.

4.2 Recruitment and ethics

The participants for the study were recruited via the snowball principle starting with close friends and family. Participants were informed about the purpose of the study (i.e. to evaluate the usability of the voting material and if it enables voters to cast their vote), as well as told that they could withdraw from the study without providing any explanation. The study was designed to be anonymous, only collecting information that the participants chose to input in the survey platform. The participants were not offered any reimbursement.

4.3 Study results

Overall, 40 participants took part in the study (while 50 received the study and voting material), of them 22 women and 18 men. The participants were aged 19-81, with the median age being 44. Most of the participants had some higher education degree (25 out of 40). Only three out of 40 participants had previously cast their vote online. Three reported to have a background in security.

Effectiveness. Overall, 95% of the participants (38 out of 40) reported being able to complete the voting procedure. Two participants did not succeed in

completing the voting procedure even after having called the support. Their device could not read the QR-Code. Note thatwe send these two the link to the survey via email to collect their feedback. One of the 38 reported contacting the study examiner at some point during the study. The issue with the election webpage could be solved with the help of the support.

Satisfaction. The SUS scores given by the 38 participants ranged from 40 to 100, with a mean score of 84.14, corresponding to the grade "B" (good) according to [5] and a standard deviation of 13.1. The mean value is close to the one reported in [17], namely, 80.9.

UEQ. The participants rated the voting system highly across all the used scales, with mean score of 2.2 ($SD = 0.76$) for dependability, 2.14 ($SD = 0.9$) for efficiency, 2.07 ($SD = 1.18$) for perspicuity and 1.45 ($SD = 1.15$) for trust, all on a scale from -3 to 3.

Feedback. Two of the authors analysed the answer provided regarding aspects participants liked and disliked (including their proposals for improvements), as well as problems they had. While more positive than negative answers were provided, in this paragraph we will focus on the negative ones as they can help to further improve the usability.

Regarding the question on *encountered problems*, 23 of the 38 participants who could cast their vote reported that they had no problem at all. Seven reported that they first had an issue with opening the webpage (e.g. due to a typo in the webpage). Two reported that they had to switch browser / device as the first setting did not enable them to scan the QR-Code. Three participants first put the voting card the opposite way, i.e. they tried to scan the return code together with the QR-Code containing the triangle.

Regarding the *positive aspects* of both the voting material and the webpage, most people mentioned at least two of the following aspects: 'easy', 'fast', 'well explained', 'clear', 'well structured', 'good usability', and 'easy language'. Often, participants were – in particular – referring in their answers to the step-by-step instruction.

Regarding the *negative aspects of the voting material*, 13 of the 38 mentioned that there is nothing they did not like. At least three times the following types of input were provided: eleven either asked specifically for further information (five particularly asked for security related information) or – from their answers – we deduced that they had misconceptions (e.g. one participant proposed to put the actual option e.g. yes on the same page of the voting card as the QR-Code is) which could be addressed by providing more information. Eight made a concrete proposal how to rephrase or extend sentences. Furthermore, three thought that the scheme is not very environmentally friendly and three, again, mentioned that typing in the URL is error prone. Regarding the *negative aspects of the webpage*, 23 of the 38 participants did not mention anything, three only mentioned again the issues with entering the URL, and two added a remark that the webpage would look more official if it would be an actual election (e.g. it would have an imprint). There was only one aspect which was mentioned at least three times: the request for more information either through a video and/or by providing

more information regarding the status in the process on the webpage. The request for more information about the security and the reason for each of the steps was also mentioned by four of the seven participants who provided input in the final-remark-question.

4.4 Result discussion and limitations

Our user study shows that it is possible to design verifiable code voting systems that achieve a high level of usability – with our proposals scoring high both in terms of effectiveness (with almost all of our study participants being able to successfully complete the voting procedure) as well as in terms of satisfaction and user experience. It is worth mentioning, that the problems with the smartphone and the camera would not have happened in a typical lab user study (as conducted in [17,22]) in which participants would have used lab equipment. Due to the remote character of our user study, we did not know which smartphone participants used and were limited wrt. testing various combinations of smartphones-OS versions - webbrowsers. Before using such an approach in the field, the webpage would need to be tested with more potential combinations, to further reduce such issues.

Beside the positive usability and user experience, our results show potential for further improvements and directions for future research: e.g. the URL for the election webpage should also be provided as QR-Code to avoid errors when typing the URL on the smartphone and it should be better explained in which way the voting card should be placed before scanning it. As it is critical that the camera of the voting device used to scan the QR codes does not capture any information that is supposed to be hidden (most notably, the correspondences between the voting cards and the voting options on them), additional studies need to be done to make sure that the voters are aware of this aspect, and that their interactions with the system do not lead to errors that might violate their vote secrecy.

We also received comments unrelated to the actual usability but which are worth to be considered as future work: Our participants wished for more transparency regarding the scheme, including information on the security of the system and explanations on why the individual steps are needed. While providing such information was out of scope for our study, developing ways to communicate it would be an important research direction for real-world elections. Such information could also explain why it is needed to have paper-based materials sent to voters and why the actual option cannot be printed on top of the QR-Code. Furthermore, as mentioned by participants, our proposal relies on availability of smartphones with cameras capable of scanning QR-Codes; while such assumption might be trivially fulfilled in Switzerland (with recent surveys showing more than 97.2% of the population in possession of a smartphone)[6], consideration of other alternatives might be useful for the applicability of code voting in other settings. A related issue is the accessibility of the system; in particular, both the original

[6] `https://de.statista.com/statistik/daten/studie/537944/umfrage/besitz-von-smartphone-bzw-tablet-in-der-schweiz/`

Swiss system as well as our improvement could require additional assistance to be used by voters with vision impairments. Alternatives to the use of QR codes, including more accessible ways to represent the codes (such as codes that are designed to incorporate error correction [7]), need to be further investigated.

Limitations. The focus of our user study was on usability and user experience aspects. We did so by sending participants their voting material home. By doing so, we increased external validity compared to previous research being conducted in the lab. However, casting the vote from home is a less controlled environment. Thus, we cannot know whether participants asked others in the household for help. We also don't know how carefully they read the materials before and while casting their vote. Different to [17, 22], we leave an evaluation of the effectiveness of the cast-as-intended verification for future work. As a consequence, we could show good usability wrt. vote casting but we do not know whether voters would be able to detect manipulations.

5 Conclusion

We proposed voting material and election webpage to enable verifiable secrecy-preserving e-voting schemes based on machine readable data aka. QR-Codes. This idea allowed us to remove the step in which the initialization code is entered by voters. We evaluated our proposal in a user study. The results of the user study show that users are able to actually use code voting in combination with the return code approach from [1]. Thus, the introduction of QR-Codes on the code sheet sent via postal service is a game changer: instead of hushing away from the voters completely non-trustworthy camera-equipped, connected computing device, we now can endorse or even mandate its usage in order to handle big data chunks correctly, without lowering usability. From a cryptographical perspective, the ability to confidentially handle big chunks of data at the voter's side has an immediate effect on the protocol design. This change enables the system to provide the voter with cryptographically sound data such as digitally signed encrypted data or zero knowledge proofs. This in turn allows to offload some of the strong trust-assumptions at the server side and paves the way for much more robust verifiable e-voting schemes using code voting. Now, it is up to the community to propose corresponding schemes knowing that it is possible to design usable voting materials and election webpages.

References

1. Verordnung der Bundeskanzlei über die elektronische Stimmabgabe (VEleS) (July 1st 2018). Die Schweizerische Bundeskanzlei (2018)
2. Acemyan, C.Z., Kortum, P., Byrne, M.D., Wallach, D.S.: Usability of voter verifiable, end-to-end voting systems: Baseline data for Helios, Prêt à Voter, and Scantegrity II. The USENIX Journal of Election Technology and Systems **2**(3), 26–56 (2014)
3. Acemyan, C.Z., Kortum, P., Byrne, M.D., Wallach, D.S.: From error to error: Why voters could not cast a ballot and verify their vote with Helios, Prêt à Voter, and

Scantegrity II. USENIX Journal of Election Technology and Systems **3**(2), 1–19 (2015)
4. Acemyan, C.Z., Kortum, P., Byrne, M.D., Wallach, D.S.: Summative Usability Assessments of STAR-Vote: A Cryptographically Secure e2e Voting System That Has Been Empirically Proven to Be Easy to Use. Human Factors pp. 1–24 (2018)
5. Bangor, A., Kortum, P., Miller, J.: Determining what individual sus scores mean: Adding an adjective rating scale. Journal of Usability Studies **4**(3), 114–123 (2009)
6. Bär, M., Henrich, C., Müller-Quade, J., Röhrich, S., Stüber, C.: Real world experiences with bingo voting and a comparison of usability. In: IAVoSS Workshop On Trustworthy Elections (WOTE) (2008)
7. Blanchard, N.K., Gabasova, L., Selker, T.: Consonant-vowel-consonants for error-free code entry. In: International Conference on Human-Computer Interaction. pp. 19–37. Springer (2019)
8. Budurushi, J., Renaud, K., Volkamer, M., Woide, M.: An investigation into the usability of electronic voting systems for complex elections. Annals of Telecommunications **71**(7-8), 309–322 (2016)
9. Chaum, D.: Surevote: technical overview. In: Proceedings of the workshop on trustworthy elections (WOTE'01) (2001)
10. Distler, V., Zollinger, M.L., Lallemand, C., Roenne, P., Ryan, P., Koenig, V.: Security–visible, yet unseen? how displaying security mechanisms impacts user experience and perceived security. In: ACM CHI. pp. 605:1–605:13 (2019)
11. Fuglerud, K.S., Røssvoll, T.H.: An evaluation of web-based voting usability and accessibility. Universal Access in the Information Society **11**(4), 359–373 (2012)
12. Gjøsteen, K., Lund, A.S.: An experiment on the security of the norwegian electronic voting protocol. Annals of Telecommunications **71**(7-8), 299–307 (2016)
13. Helbach, J., Schwenk, J.: Secure internet voting with code sheets. In: E-Voting and Identity. pp. 166–177. Springer (2007)
14. Joaquim, R., Ribeiro, C., Ferreira, P.: Veryvote: A voter verifiable code voting system. In: E-Voting and Identity. pp. 106–121. Springer (2009)
15. Karayumak, F., Olembo, M.M., Kauer, M., Volkamer, M.: Usability Analysis of Helios-An Open Source Verifiable Remote Electronic Voting System. In: Proceedings of the 2011 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections. EVT/WOTE'11, USENIX Association (2011)
16. Kulyk, O., Neumann, S., Budurushi, J., Volkamer, M.: Nothing comes for free: How much usability can you sacrifice for security? IEEE Security & Privacy **15**(3), 24–29 (2017)
17. Kulyk, O., Volkamer, M., Müller, M., Renaud, K.: Towards improving the efficacy of code-based verification in internet voting. In: FC: VOTING Workshop. pp. 291–309. Springer (2020)
18. MacNamara, D., Gibson, P., Oakley, K.: A preliminary study on a DualVote and Prêt à Voter hybrid system. In: CeDEM. p. 77 (2012)
19. MacNamara, D., Scully, T., Gibson, P.: Dualvote addressing usability and verifiability issues in electronic voting systems (2011)
20. Marky, K., Kulyk, O., Renaud, K., Volkamer, M.: What Did I Really Vote For? In: ACM CHI. p. 176 (2018)
21. Marky, K., Schmitz, M., Lange, F., Mühlhäuser, M.: Usability of Code Voting Modalities. In: ACM CHI (2019)
22. Marky, K., Zimmermann, V., Funk, M., Daubert, J., Bleck, K., Mühlhäuser, M.: Improving the Usability and UX of the Swiss Internet Voting Interface. In: ACM CHI (2020)

23. MARKY, K., ZOLLINGER, M.L., ROENNE, P., RYAN, P.Y., GRUBE, T., KUNZE, K.: Investigating usability and user experience of individually verifiable internet voting schemes. ACM Trans. Comput.-Hum. Interact **28**(5) (2021)
24. Neumann, S., Feier, C., Sahin, P., Fach, S.: Pretty understandable democracy 2.0 (2014), `https://eprint.iacr.org/2014/625.pdf`, [Online, May 14th 2021]
25. Olembo, M.M., Renaud, K., Bartsch, S., Volkamer, M.: Voter, what message will motivate you to verify your vote. In: USEC. Internet Society (2014)
26. Olembo, M.M., Bartsch, S., Volkamer, M.: Mental Models of Verifiability in Voting. In: E-Voting and Identity. pp. 142–155. Springer (2013)
27. Oostveen, A.M., Van den Besselaar, P.: Users' experiences with e-voting: A comparative case study. Journal of Electronic Governance **2**(4) (2009)
28. Oppliger, R.: How to address the secure platform problem for remote internet voting (2002), `http://pubs.esecurity.ch/sis_2002.pdf`, [Online, May 14th 2021]
29. Schneider, S., Llewellyn, M., Culnane, C., Heather, J., Srinivasan, S., Xia, Z.: Focus group views on prêt à voter 1.0. In: REVOTE,. pp. 56–65. IEEE (2011)
30. Schrepp, M., Thomaschewski, J.: Design and validation of a framework for the creation of user experience questionnaires. International Journal of Interactive Multimedia & Artificial Intelligence **5**(7) (2019)
31. Weber, J., Hengartner, U.: Usability study of the open audit voting system Helios (2009)
32. Wikipedia: Electronic voting in estonia, `https://en.wikipedia.org/w/?title=Electronic_voting_in_Estonia&oldid=1012067015`, [Online, May 14th 2021]
33. Winckler, M., Bernhaupt, R., Palanque, P., Lundin, D., Leach, K., Ryan, P., Alberdi, E., Strigini, L.: Assessing the Usability of Open Verifiable E-Voting Systems: a Trial with the System Prêt à Voter. In: ICE-GOV. pp. 281–296 (2009)